

Simulacre prova 1ª part UF1: Protocol HTTP/S i implementació servidor Apache

HTTP i REST

1.-a) Després de consultar (a <https://www.boredapi.com/documentation>) la documentació de l'API "Bored", digues com faries servir la comanda *curl* contra el seu servidor per mostrar al terminal només (és a dir, hauràs de fer neteja amb *jq*) la descripció d'una activitat aleatòria que sigui de franc i només per una sola persona.

NOTA: Un bon tutorial de *jq* el pots trobar aquí: <http://www.compciv.org/recipes/cli/jq-for-parsing-json>, però recorda que tens disponible un pdf en exclusiva on es mostren les seves opcions més importants

aII) Digues com faries servir la comanda *ncat* per obtenir la mateixa resposta que amb el *curl* de l'apartat anterior i prova-ho.

NOTA: Probablement caldrà que facis servir el paràmetre *-C* del *ncat*. Aquest paràmetre fa que els "Enter" que pulsem interactivament mentre estem construint la petició no siguin interpretats com "\n" (que és el que passa per defecte) sinó com "\r\n", que és la combinació de caràcter estàndar a l'HTTP per delimitar línies.

b) Després de consultar (a <https://dog.ceo/dog-api/documentation>) la documentació de l'API "Dog", escriu, en una màquina amb escriptori Gnome instal·lat, la combinació de comandes *eog*, *curl -s* i *jq -r* necessària (on segurament hi intervindrà alguna canonada *-'|'* i alguna substitució de comandes *"\$(...)"*) per mostrar a pantalla la foto d'un gos qualsevol triat a l'atzar.

bII) Explica què fan cadascuna de les comandes següents, indicant el per què de cadascun dels paràmetres que tenen: *eog https:\$(curl -s http://www.google.com/doodles#oodles/archive | grep -A5 "latest-doodle on" | grep "img src" | cut -f2 -d"\")*

NOTA: Per fer servir les cometes com a caràcter separador a la comanda *cut*, el paràmetre *-d* s'ha d'escriure tal com es veu, així: *-d "\"* per tal "d'escapar" la cometa

c) Després de consultar (a <https://ip-api.com/docs>) la documentació de l'API "IP-API", escriu la comanda *curl* correcta per obtenir en format JSON la ciutat i país on s'ubica l'adreça IP 1.1.1.1 I també la comanda *curl* per obtenir la mateixa informació però en format CSV

Ntfy.sh (<https://ntfy.sh>) és un servei online gratuït que funciona com un "broker" de missatges: a ell se li poden enviar via peticions HTTP de tipus POST missatges d'un determinat tema (i amb un determinat contingut) als quals altres clients poden subscriure's (via una petició GET permanent) per, a partir de llavors, rebre'ls de forma síncrona en el moment que el client publicador en faci un enviament al "broker".

NOTA: El "broker" no cal que sigui "ntfy.sh" sinó que podem implementar-ne un "self-hosted" en els nostres propis servidors seguint els passos descrits a <https://docs.ntfy.sh/install/>

d) Després de consultar (a <https://docs.ntfy.sh/publish>) la documentació de l'API "Ntfy" relativa a l'enviament de missatges al seu "broker" online, escriu la comanda *curl* correcta per enviar un missatge amb el tema que vulguis però amb el contingut "S.O.S estem en un vaixell fantasma".

dII) Després de consultar (a <https://docs.ntfy.sh/subscribe/api>) la documentació de l'API "Ntfy" relativa a la subscripció de missatges d'un determinat tema al seu "broker" online, escriu la comanda *curl* correcta per anar mostrant en temps real a la pantalla tots els missatges que vagi publicant (cada cop que s'executi) el client emissor de l'apartat anterior.

NOTA: A l'igual que per fer enviaments, Ntfy proporciona diferents formes de subscriure's a un tema i, per tant, de visualitzar els missatges corresponents un cop rebut que no són amb la comanda *curl* (per exemple, via app de mòbil, via navegador integrat amb el sistema de notificacions de l'escriptori, via una aplicació de terminal específica, etc)

2.-a) Consulta *man curl* (i/o els apunts de classe) i digues amb les teves pròpies paraules:

- ¿Per a què serveix el seu paràmetre *-F* i quin és el mètode HTTP que fa servir?
- ¿Quin és el valor de la capçalera "Content-Type" que genera el paràmetre *-F*?
- ¿Quina és la diferència entre els símbols @ o < en indicar el valor del paràmetre *-F*?
- ¿I si es combina algun dels dos símbols anteriors amb el símbol - ?
- ¿Per a què serveix el paràmetre *-T* i quin és el mètode HTTP que fa servir?

b) A partir de la informació anterior, dedueix què fan les següents comandes (les quals fan servir diferents serveis online similars) i digues quina és la funció de cadascun dels paràmetres utilitzats (i el del seu valor indicat respectiu).

```
ps -e | curl -F "clbin=<-" https://clbin.com      curl -F "file=@a.yml" http://0x0.st
cat file.txt | curl -F "f:1=<-" http://ix.io      curl -F "f:1=@file.txt" http://ix.io
```

Apache

3.-Implementa, en una màquina virtual amb la seva tarja en mode adaptador pont, un servidor Apache que tingui les següents característiques:

a) Fes que ofereixi dos "virtualhosts", un amb el nom de "www.marca.com" i un altre amb el nom de "www.sport.es". El primer haurà de guardar un registre de les IPs dels clients que li han connectat dins de l'arxiu "/var/log/marca.txt" i el segon haurà de tenir una pàgina personalitzada per l'error 404. Ambdós hauran d'oferir un DocumentRoot però només el primer hi allotjarà un arxiu anomenat "index.html" (amb el contingut "Hola, sóc Marca"); el segon no tindrà cap contingut dins del seu DocumentRoot

NOTA: Recorda d'alterar la resolució de noms del client per a què tant "www.marca.com" com "www.sport.es" apuntin a l'adreça IP del servidor Apache

b) Després de copiar dins del DocumentRoot del "virtualhost" "www.sport.es" algun fitxer qualsevol, fes que en visitar els clients aquest DocumentRoot se'ls mostri un índex de continguts, el qual haurà de tenir un títol personalitzat.

c) Després de crear dins del DocumentRoot del "virtualhost" "www.sport.es" una carpeta anomenada "secret" amb un sol fitxer al seu interior (anomenat "contrasenya.txt" i amb un contingut qualsevol), fes que en visitar els clients aquesta carpeta es produeixi un procés d'autenticació HTTP de tipus Digest que només permeti accedir-hi amb un usuari anomenat "miguelito" amb contrasenya "1234".

d) Fes que el "virtualhost" "www.marca.com" tingui definit un àlies "/pepe" que apunti a la carpeta "/etc" del sistema i que, en visitar els clients la URL corresponent ("www.marca.com/pepe"), se'ls mostri també un índex de continguts.

dII) Fes que l'àlies anterior només estigui disponible per un sol client amb una determinada IP (la que tingui la màquina que estiguis fent servir com a client en aquest exercici) i, a més, només entre les 16h i 20h

e) Fes que en visitar els clients la URL "www.marca.com/manolo" siguin redireccionats automàticament a "www.mundodeportivo.com"

f) Fes que el mòdul "userdir" estigui activat al "virtualhost" "www.marca.com" i fes-lo servir

g) Fes que el mòdul "status" estigui activat al "virtualhost" "www.sport.es" i fes-lo servir

4.-a) En la mateixa màquina virtual on s'estigui executant el servidor Apache de l'exercici anterior, crea-li un certificat TLS que estigui signat per la clau privada d'una CA que seràs tu mateix (és a dir, repeteix l'exercici 1 del PDF sobre OpenSSL)

b) Configura el servidor Apache per tal què ofereixi els dos "virtualhosts" que ara mateix té implementats només via HTTPS gràcies al certificat TLS (i la clau privada) creats a l'apartat anterior.